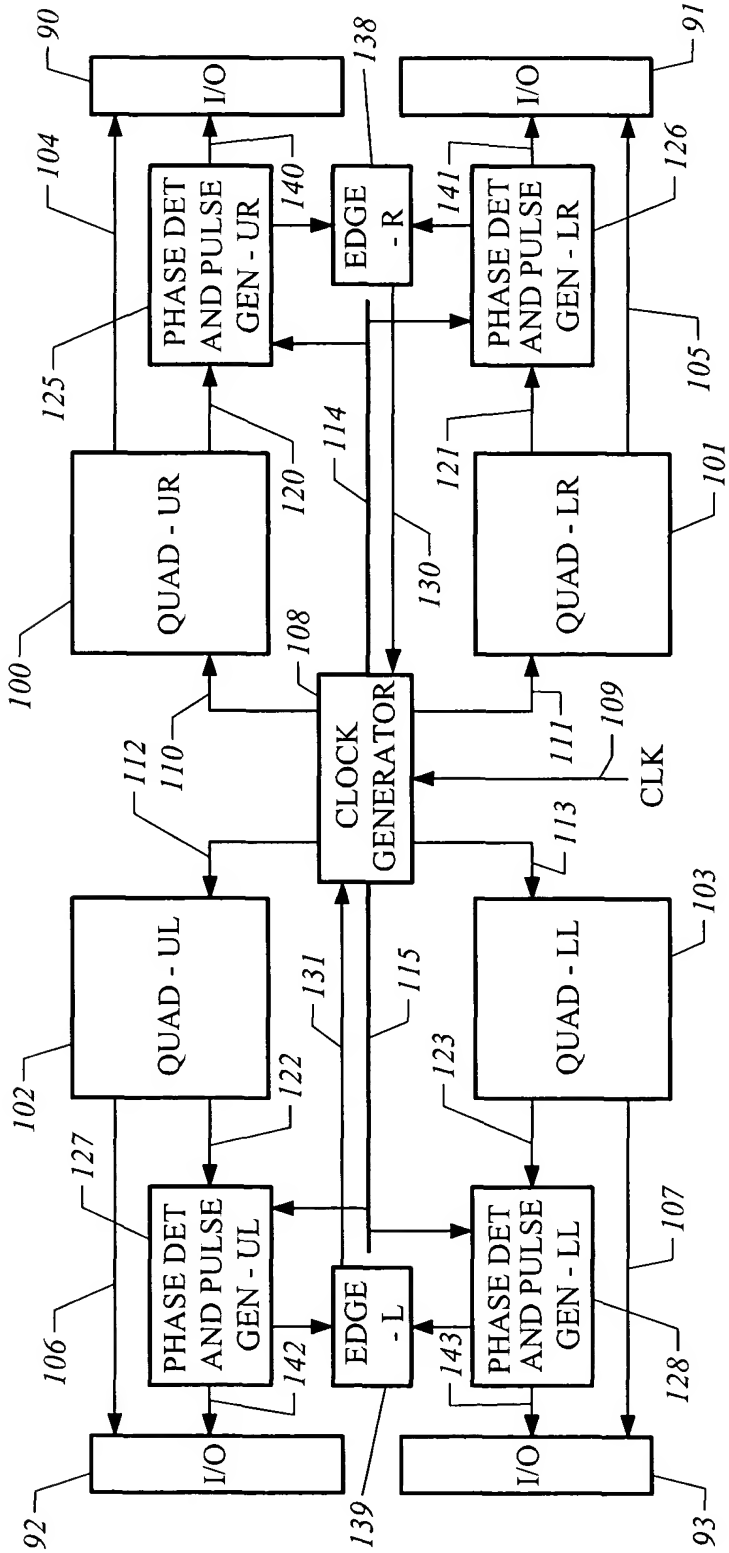

90
I/O
138
91
I/O
104
PHASE DET AND PULSE GEN - UR
140
EDGE - R
141
PHASE DET AND PULSE GEN - LR
126
125
105
114
120
121
QUAD - UR
QUAD - LR
100
101
130
108
CLOCK GENERATOR
109
CLK
112
110
111
113
QUAD - UL
QUAD - LL
102
103
131
115
123
122
127
PHASE DET AND PULSE GEN - UL
EDGE - L
PHASE DET AND PULSE GEN - LL
107
106
142
143
128
I/O
I/O
92
139
93

FIG. 1

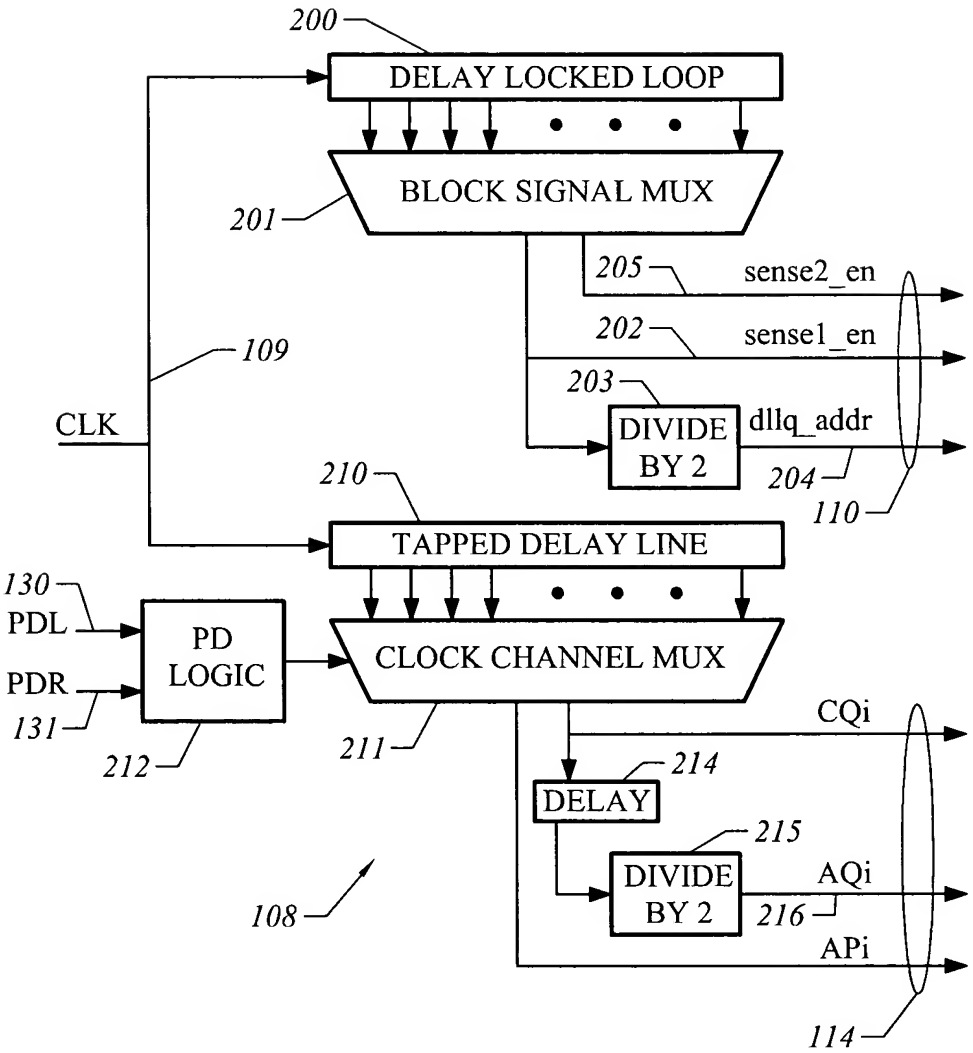

200
DELAY LOCKED LOOP
BLOCK SIGNAL MUX
201
205
sense2_en
202
sense1_en
203
DIVIDE BY 2
dllq_addr
204
110
109
CLK
210
TAPPED DELAY LINE
130
PDL
PDR
131
PD LOGIC
212
CLOCK CHANNEL MUX
211
CQi
214
DELAY
215
DIVIDE BY 2
AQi
216
APi
114
108

FIG. 2

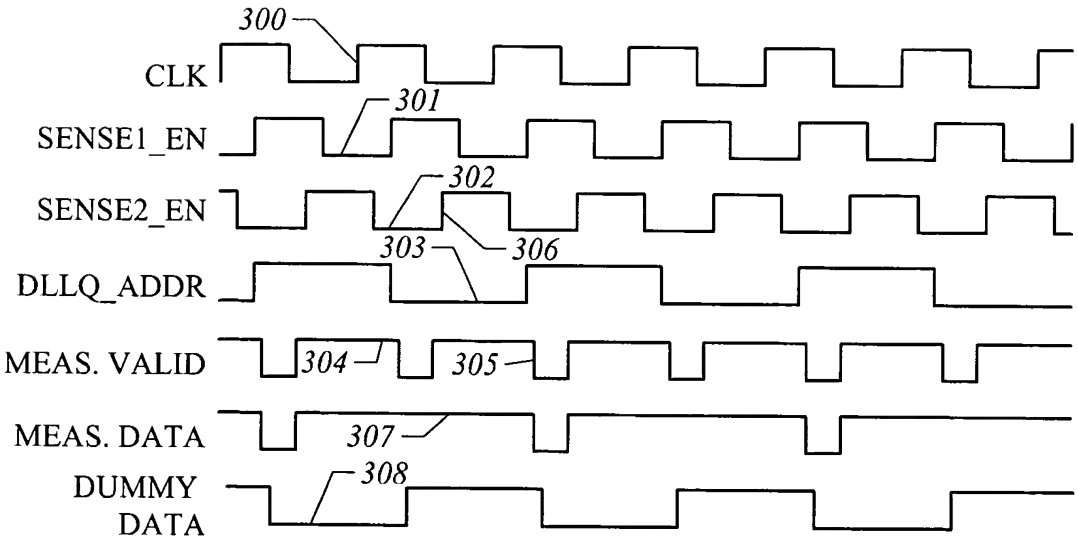

300
CLK
301
SENSE1_EN
302
SENSE2_EN
303
306
DLLQ_ADDR
MEAS. VALID
304
305
MEAS. DATA
307
308
DUMMY DATA

FIG. 3

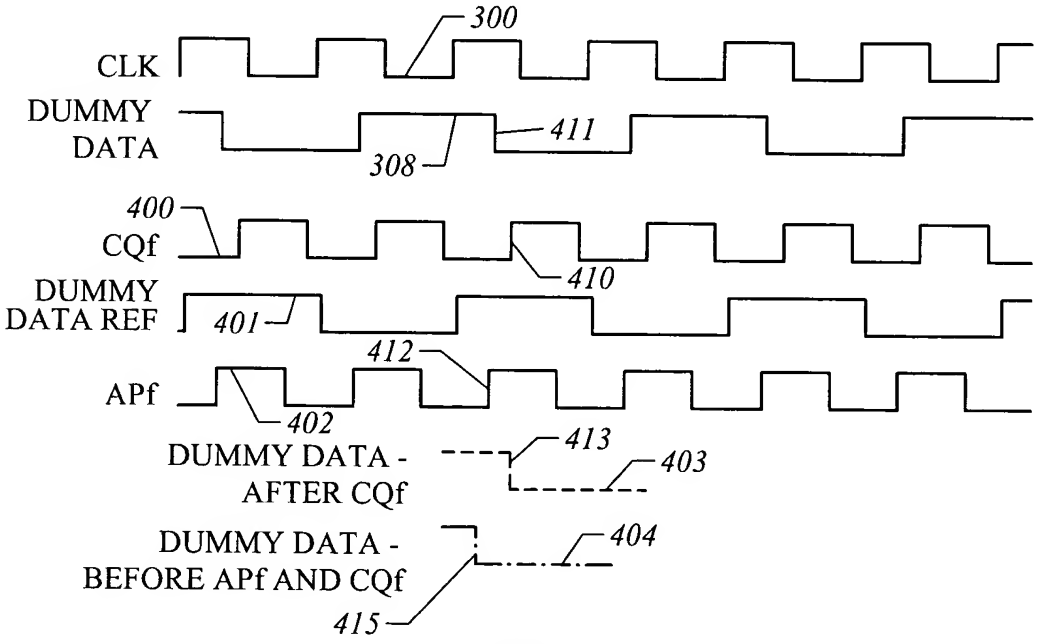

300
CLK
DUMMY DATA
411
308
400
CQf
410
DUMMY DATA REF
401
412
APf
402
DUMMY DATA - AFTER CQf
413
403
DUMMY DATA - BEFORE APf AND CQf
404
415

FIG. 4

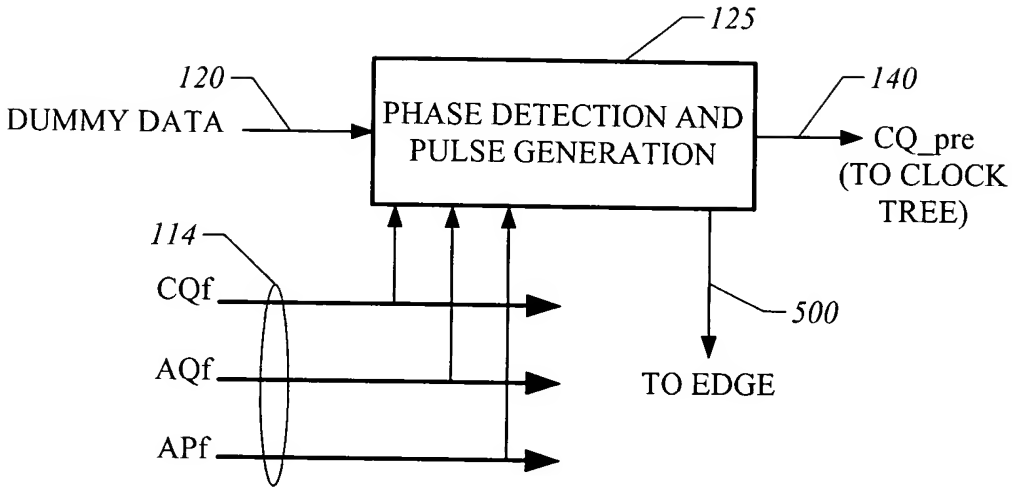

125
120
DUMMY DATA
PHASE DETECTION AND PULSE GENERATION
140
CQ_pre (TO CLOCK TREE)
114
CQf
AQf
APf
500
TO EDGE

FIG. 5

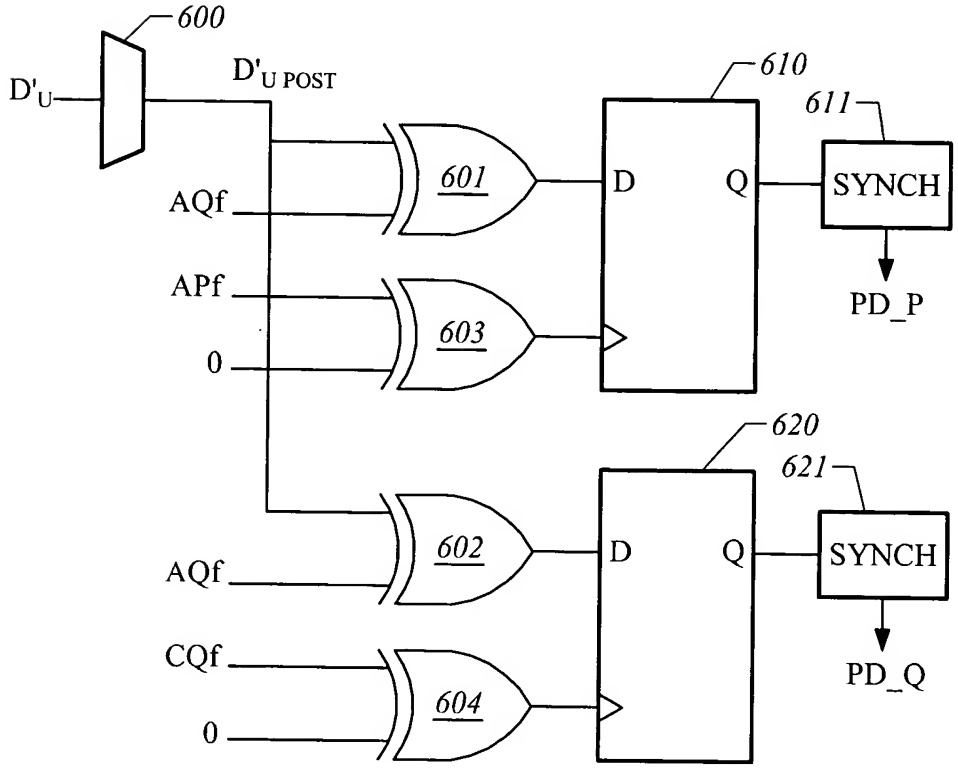
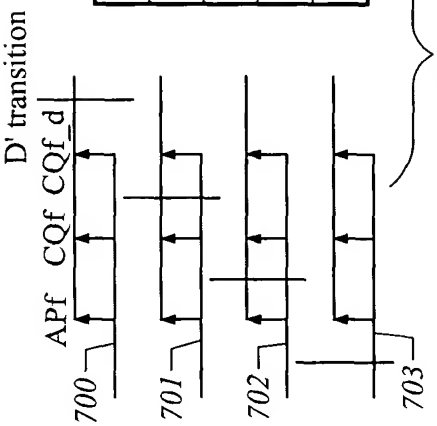

600
D'U
D'U POST
610
611
AQf
601
D
Q
SYNCH
PD_P
APf
603
0
620
621
AQf
602
D
Q
SYNCH
PD_Q
CQf
604
0

FIG. 6



PD_P	PD_Q	Decision (pdraw<2:0>=PD_P,PD_Q,1)
0	0	Increase the delay (001)
0	0	Increase the delay (001)
0	1	Keep the delay constant (011)
1	1	Decrease the delay (111)

FIG. 7

PDX<0>=NAND(pdraw_u<1>, pdraw_1<1>)
PDX<1>=AND(pdraw_u<2>, pdraw_1<2>)

PDR<1:0>	PDL<1:0>	Decision
Increase (01)	X	Increase the delay
X	Increase (01)	Increase the delay
Decrease (10)	Decrease (10)	Decrease the delay (Only if the request is repeated for n cycles, n=k* #(scanned banks in one quadrant)
All other cases		Keep the delay constant

FIG. 8

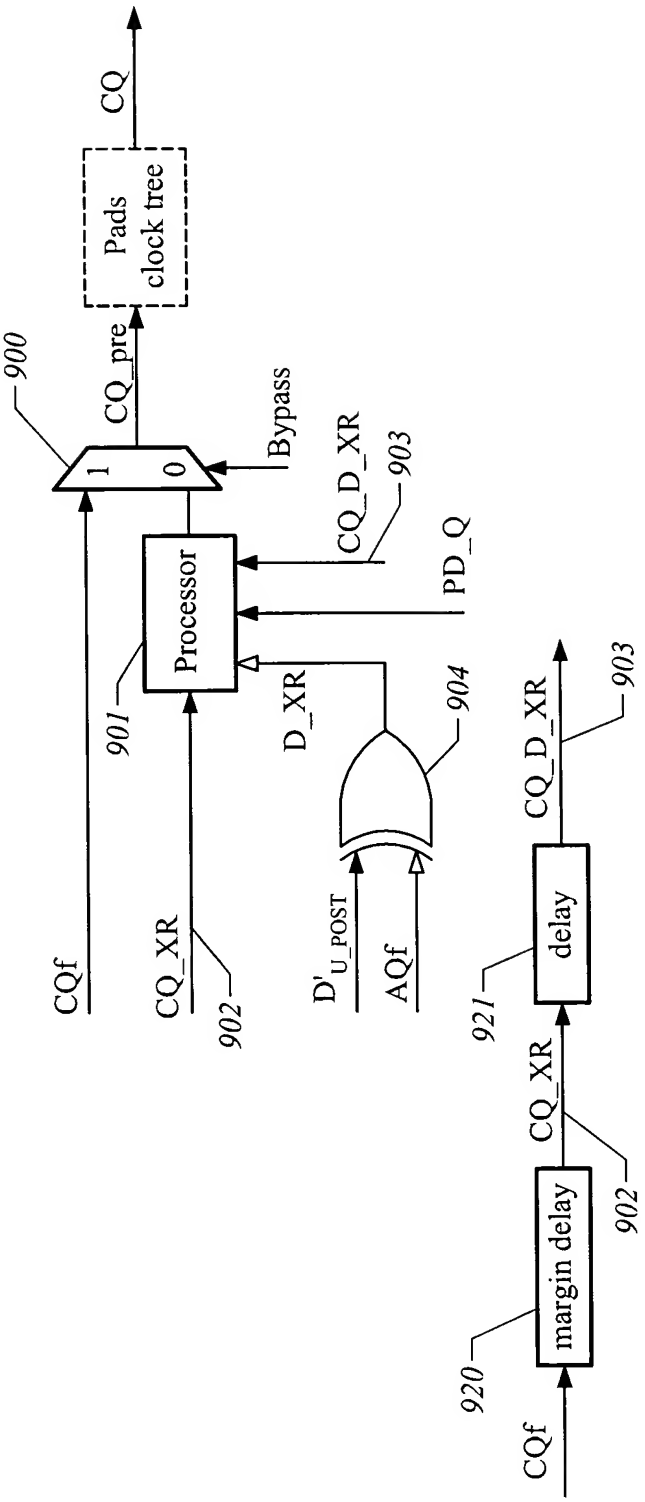

CQ
Pads clock tree
CQ_pre
900
1
0
Bypass
CQ_D_XR
903
PD_Q
Processor
901
D_XR
904
CQf
CQ_XR
902
D'U_POST
AQf
CQ_D_XR
903
delay
921
CQ_XR
902
margin delay
920
CQf

FIG. 9

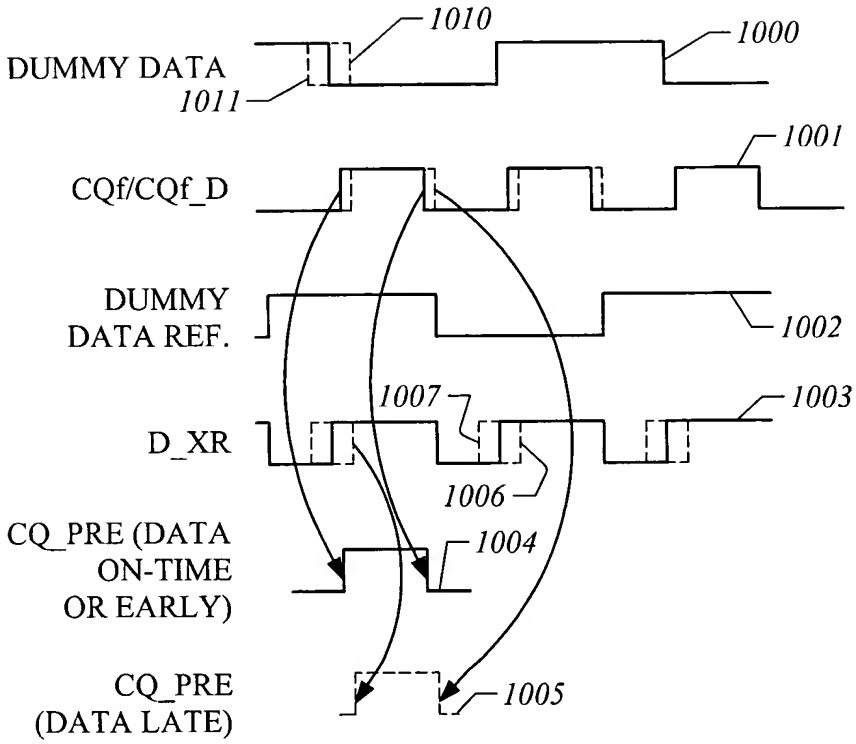

1010
1000
DUMMY DATA
1011
1001
CQf/CQf_D
DUMMY
DATA REF.
1002
1007
1003
D_XR
1006
CQ_PRE (DATA
ON-TIME
OR EARLY)
1004
CQ_PRE
(DATA LATE)
1005

FIG. 10

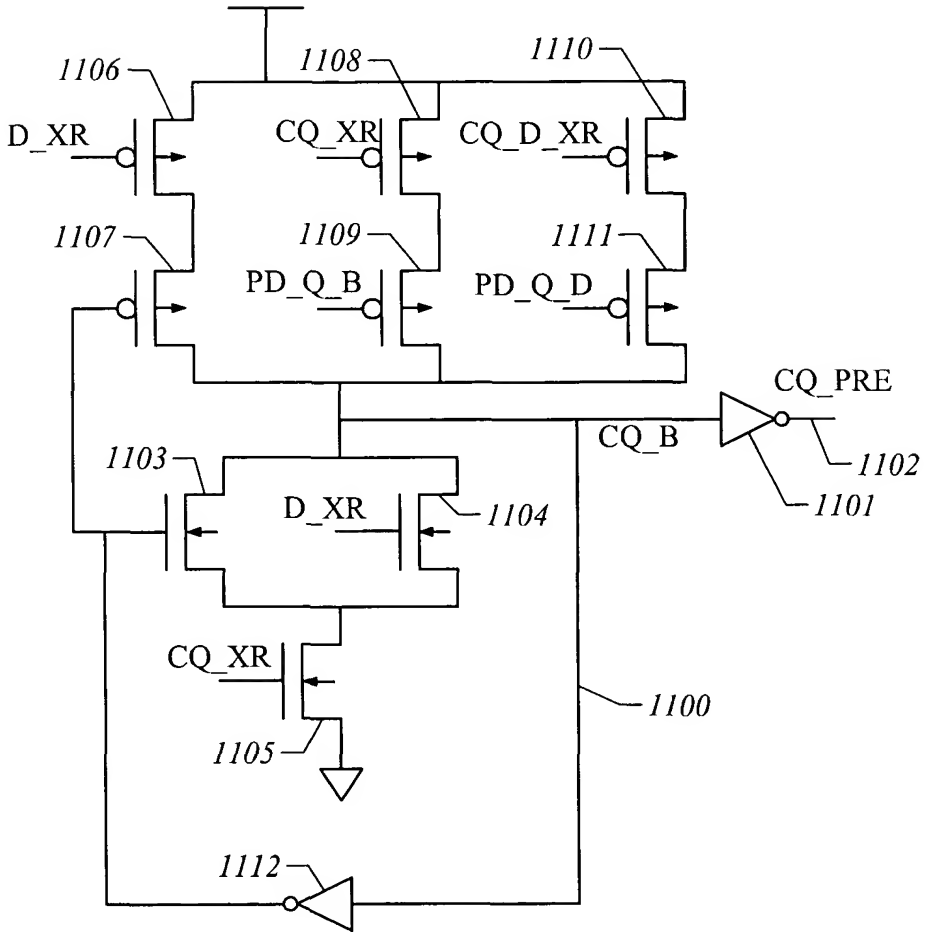

1106
1108
1110
D_XR
CQ_XR
CQ_D_XR
1107
1109
1111
PD_Q_B
PD_Q_D
CQ_PRE
CQ_B
1102
1101
1103
D_XR
1104
CQ_XR
1100
1105
1112

FIG. 11

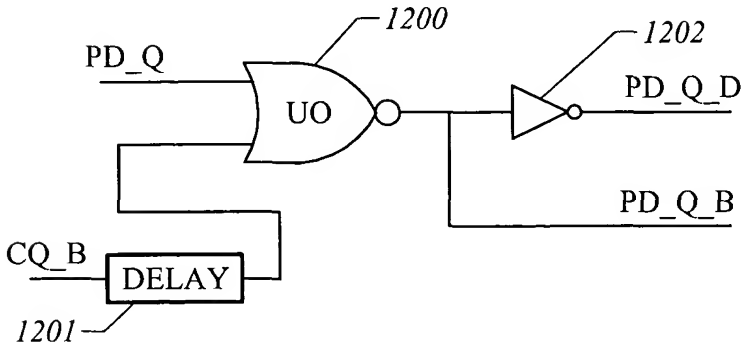

1200
1202
PD_Q
UO
PD_Q_D
PD_Q_B
CQ_B
DELAY
1201

FIG. 12

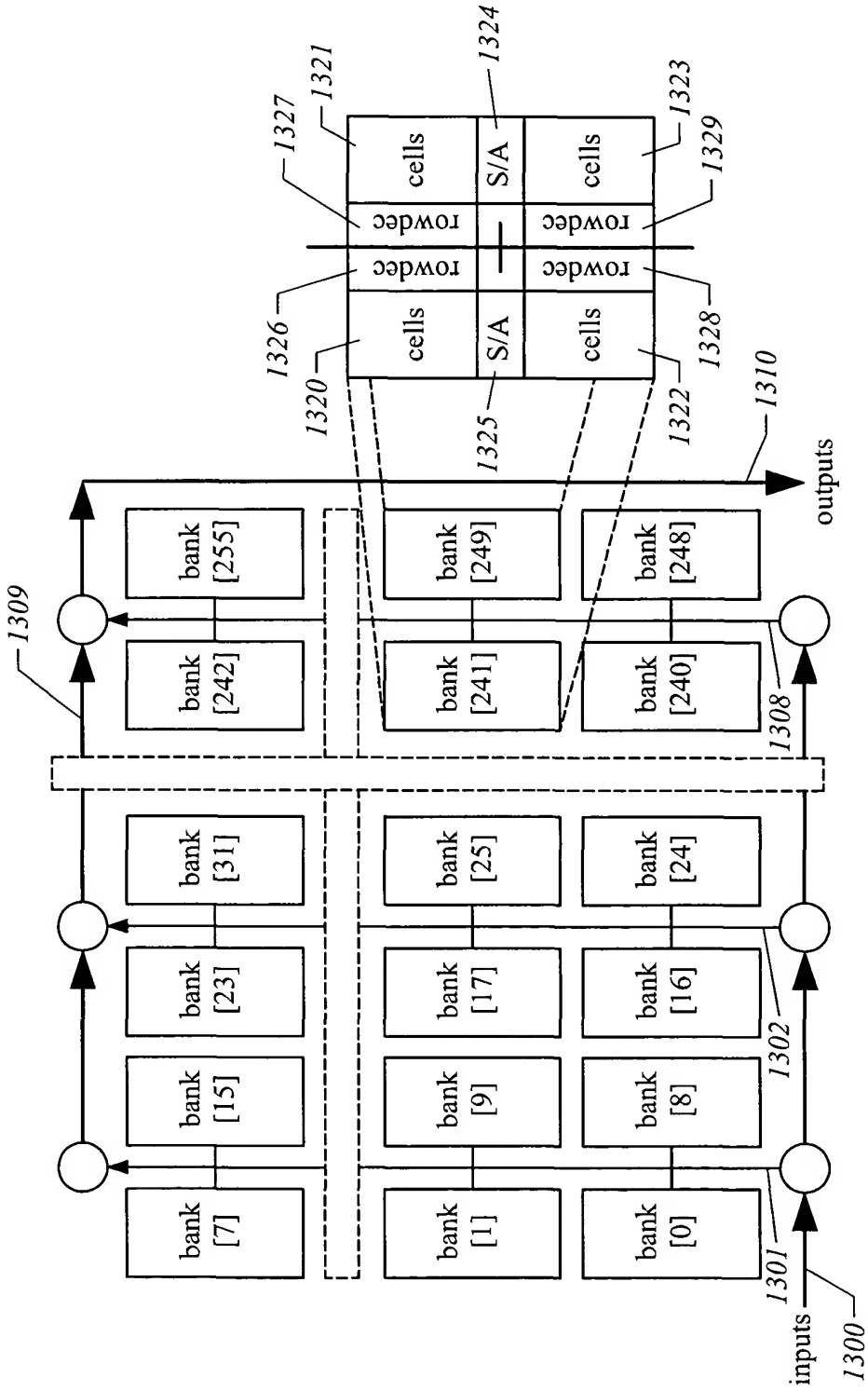

1324
1321
1327
1323
1329
cells
S/A
cells
rowdec
rowdec
rowdec
rowdec
cells
S/A
cells
1326
1328
1320
1322
1325
1310
outputs
1309
bank [255]
bank [249]
bank [248]
bank [242]
bank [241]
bank [240]
1308
bank [31]
bank [25]
bank [24]
bank [23]
bank [17]
bank [16]
1302
bank [15]
bank [9]
bank [8]
bank [7]
bank [1]
bank [0]
1301
inputs
1300

FIG. 13

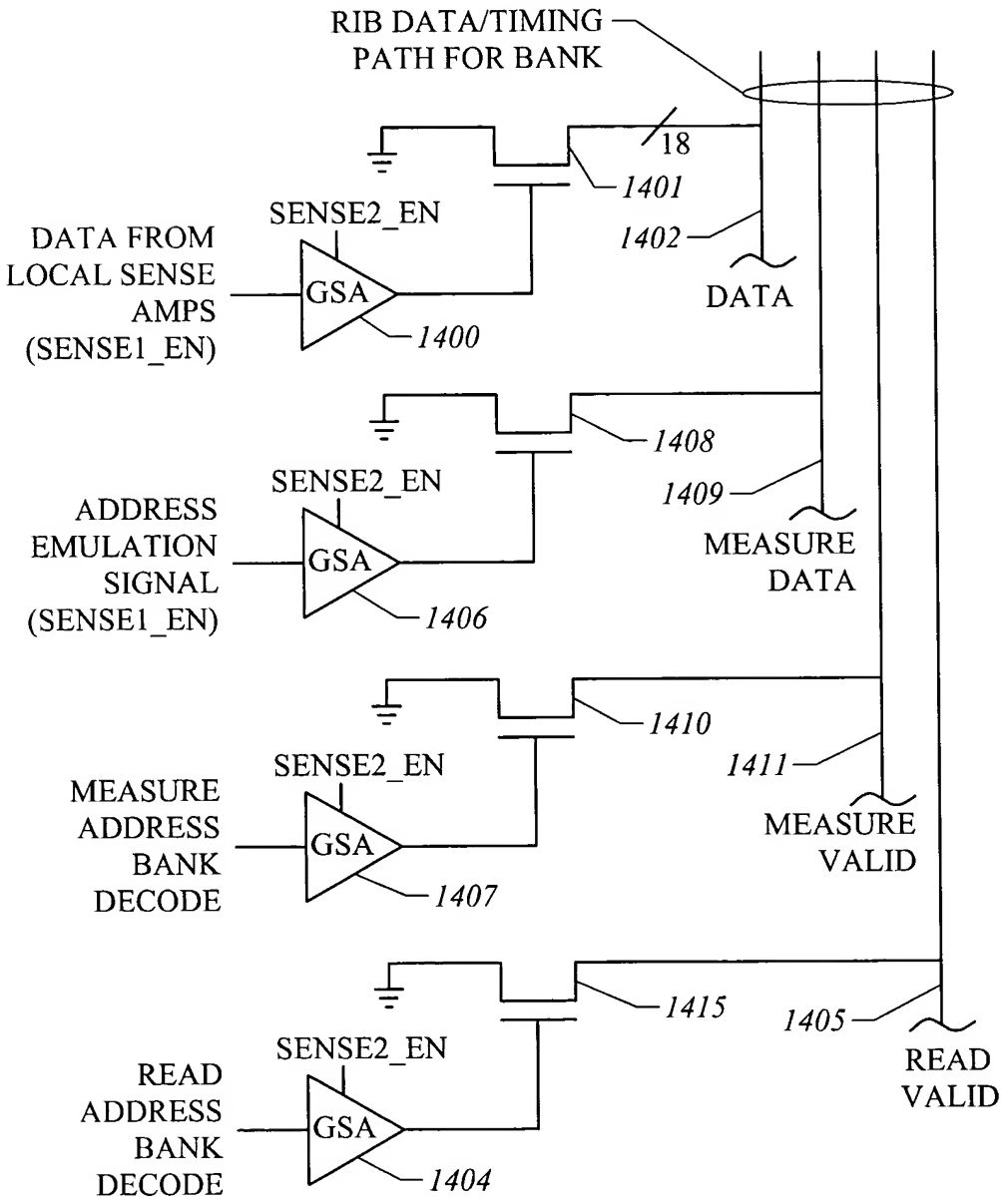

RIB DATA/TIMING PATH FOR BANK
18
1401
1402
DATA
SENSE2_EN
DATA FROM LOCAL SENSE AMPS (SENSE1_EN)
GSA
1400
1408
1409
MEASURE DATA
SENSE2_EN
ADDRESS EMULATION SIGNAL (SENSE1_EN)
GSA
1406
1410
1411
MEASURE VALID
SENSE2_EN
MEASURE ADDRESS BANK DECODE
GSA
1407
1415
1405
READ VALID
SENSE2_EN
READ ADDRESS BANK DECODE
GSA
1404

FIG. 14

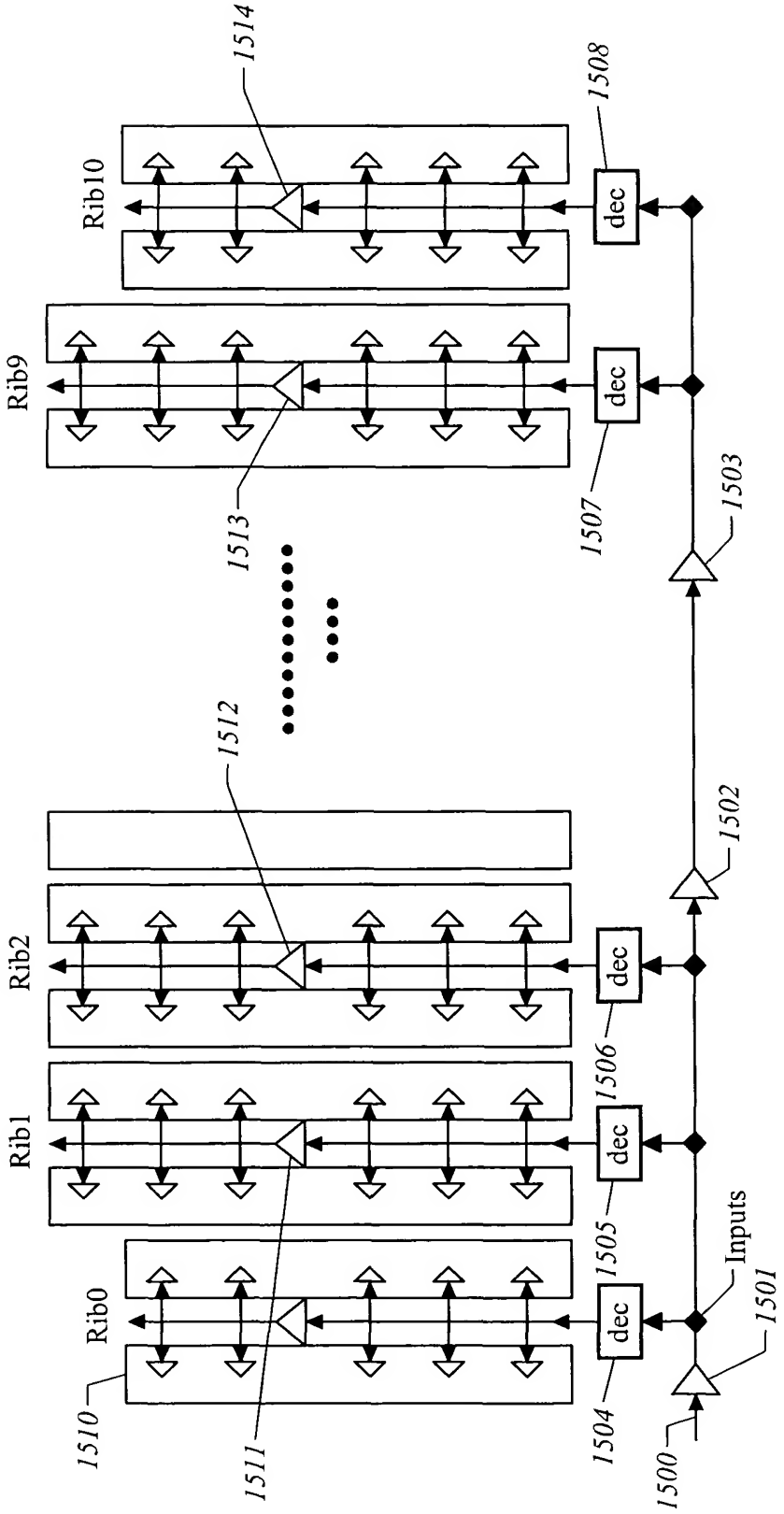

1514
1508
Rib10
dec
Rib9
dec
1513
1507
1503
1512
1502
Rib2
dec
1506
Rib1
dec
1505
Inputs
1501
Rib0
dec
1504
1510
1511
1500

FIG. 15

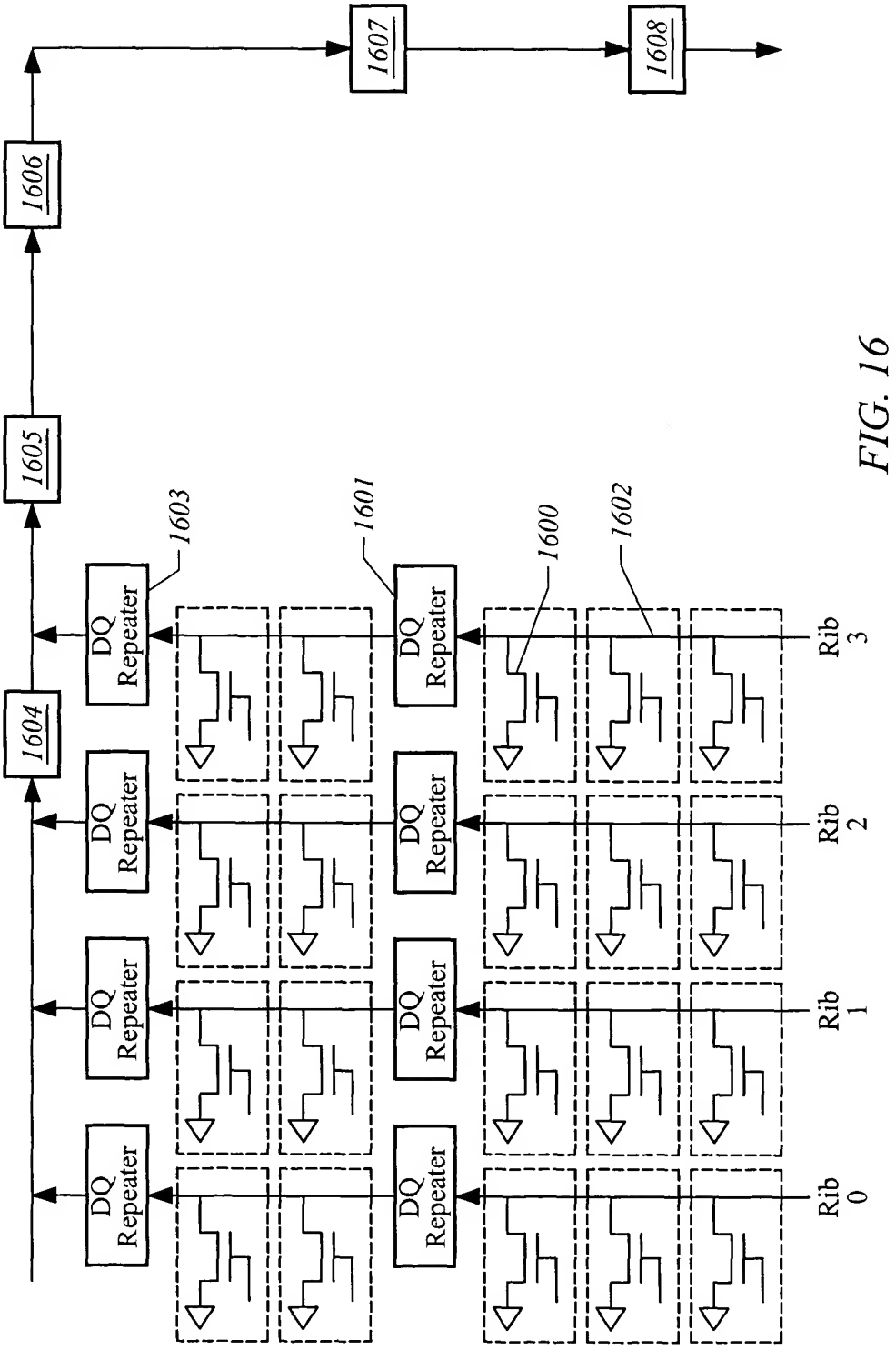

1607
1608
1606
1605
1604
1603
1601
1600
1602
DQ Repeater
Rib 3
Rib 2
Rib 1
Rib 0

FIG. 16

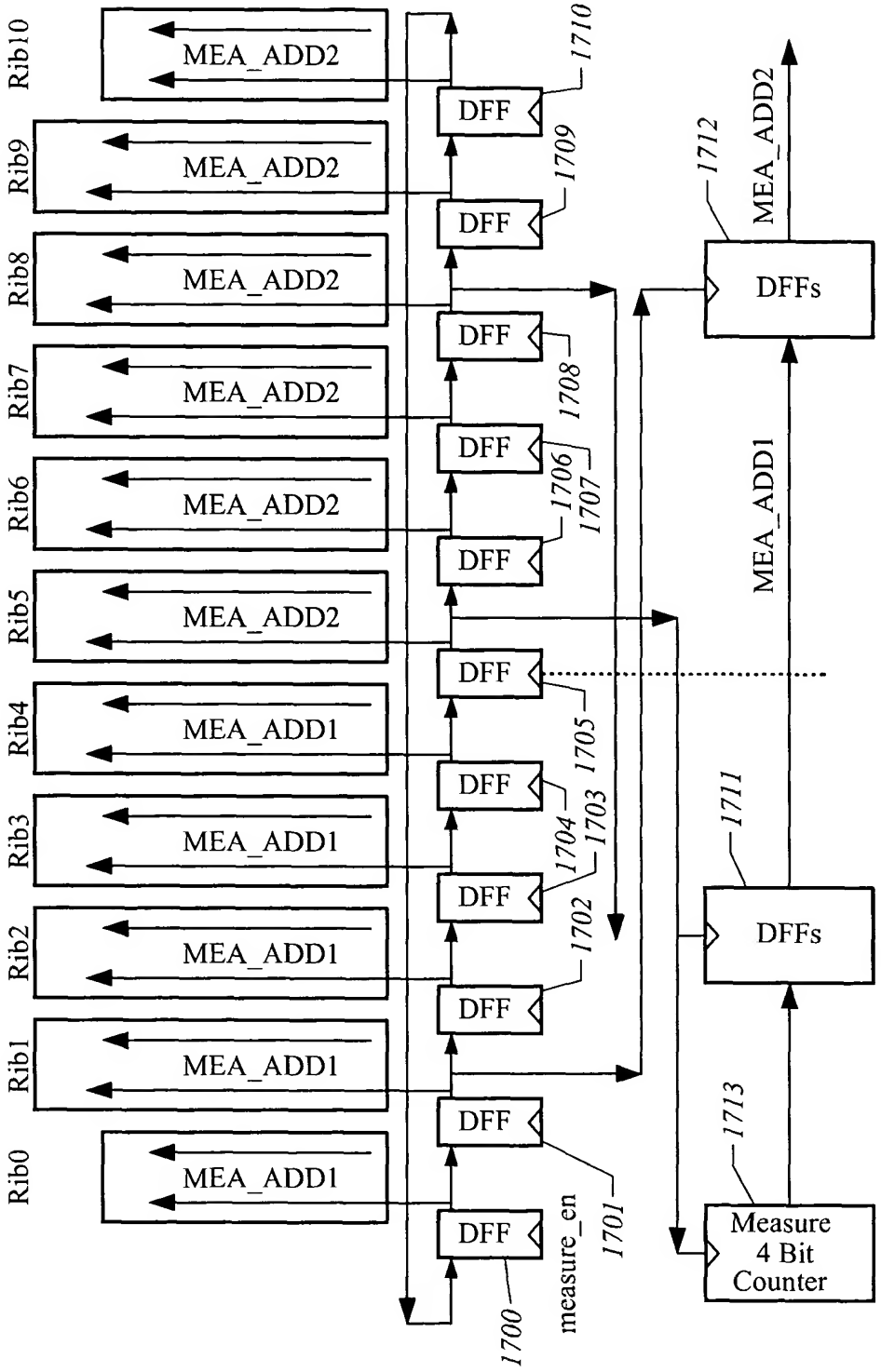

Rib10
Rib9
Rib8
Rib7
Rib6
Rib5
Rib4
Rib3
Rib2
Rib1
Rib0
MEA_ADD2
MEA_ADD1
DFF
1710
1709
1708
1706
1707
1705
1704
1703
1702
1701
1700
measure_en
1712
1711
1713
DFFs
MEA_ADD1
MEA_ADD2
Measure
4 Bit
Counter

FIG. 17

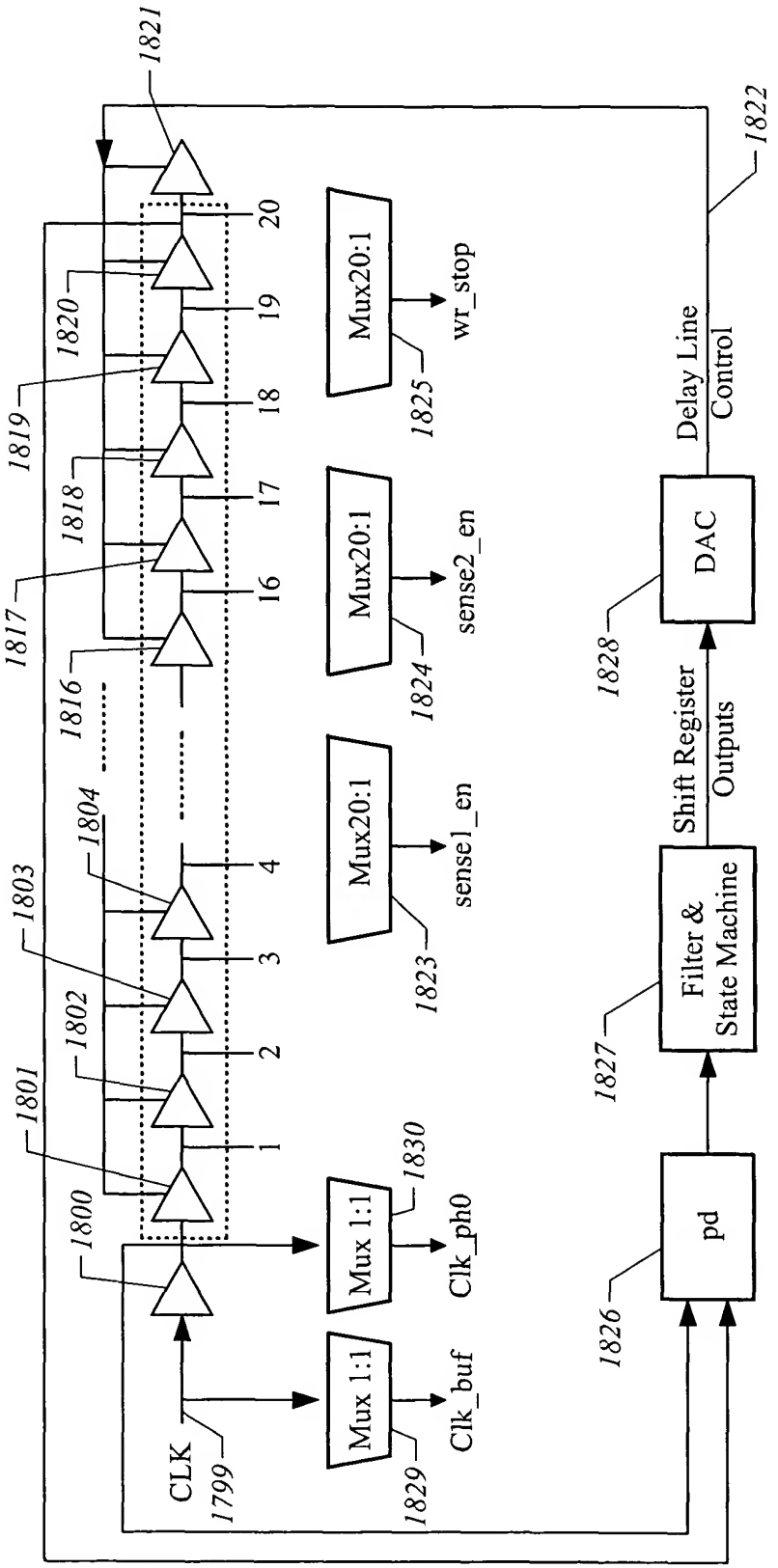

1821
1822
1819
1820
1818
1817
1816
1804
1803
1802
1801
1800
1799
CLK
Clk_buf
Clk_ph0
1829
1830
Mux 1:1
Mux 1:1
1823
1824
1825
Mux20:1
Mux20:1
Mux20:1
sense1_en
sense2_en
wr_stop
1826
1827
1828
pd
Filter & State Machine
DAC
Shift Register
Outputs
Delay Line
Control
1
2
3
4
16
17
18
19
20

FIG. 18

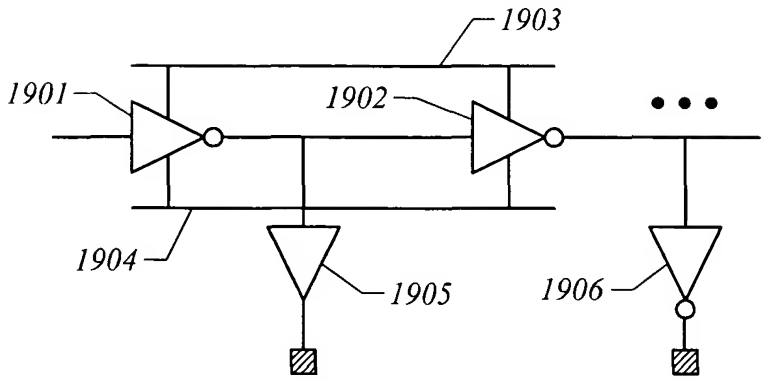

1903
1901
1902
1904
1905
1906

FIG. 19

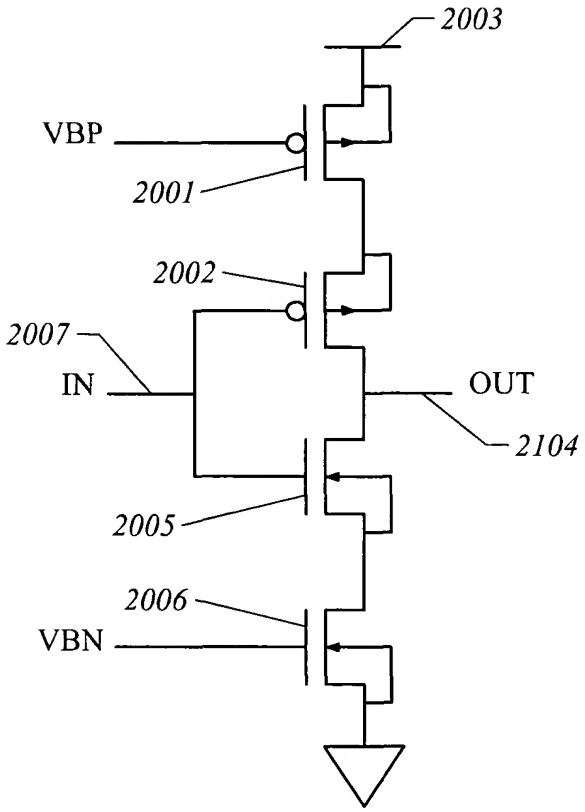

2003
VBP
2001
2002
2007
IN
OUT
2104
2005
2006
VBN

FIG. 20

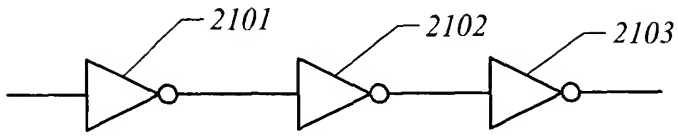


FIG. 21

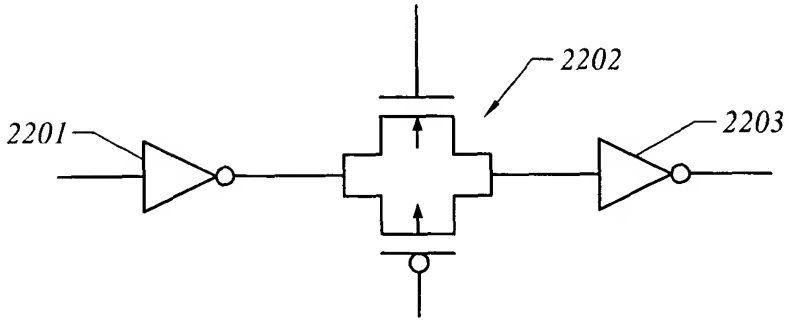


FIG. 22

Lower Control Bits mux_s<4:0> mux_sp<4:0>					Higher Control Bits mux_sh<3:0> mux_shp<3:0>				cqf, aqf apf
4	3	2	1	0	3	2	1	0	Clock phase
X	X	X	X	X	0	0	0	0	Z
0	0	0	0	0	X	X	X	X	Z
0	0	0	0	1	0	0	0	1	In0
0	0	0	1	0	0	0	0	1	In1
0	0	1	0	0	0	0	0	1	In2
0	1	0	0	0	0	0	0	1	In3
1	0	0	0	0	0	0	0	1	In4
0	0	0	0	1	0	0	1	0	In5
0	0	0	1	0	0	0	1	0	In6
0	0	1	0	0	0	0	1	0	In7
0	1	0	0	0	0	0	1	0	In8
1	0	0	0	0	0	0	1	0	In9
0	0	0	0	1	0	1	0	0	In10
0	0	0	1	0	0	1	0	0	In11
0	0	1	0	0	0	1	0	0	In12
0	1	0	0	0	0	1	0	0	In13
1	0	0	0	0	0	1	0	0	In14
0	0	0	0	1	1	0	0	0	In15
0	0	0	1	0	1	0	0	0	In16
0	0	1	0	0	1	0	0	0	In17
0	1	0	0	0	1	0	0	0	In18
1	0	0	0	0	1	0	0	0	In19

FIG. 23

Path1:			
mux_s<4:0>	mux_sh<3:0>	Clock Phase	
10000	0001	In4	
↓	↓	↓	
10000	0010 (Transition state)	In9	
↓	↓	↓	
00001	0010	In5	

Path 2:			
mux_s<4:0>	mux_sh<3:0>	Clock Phase	
10000	0001	In4	
↓	↓	↓	
00001	0001 (Transition state)	In0	
↓	↓	↓	
00001	0010	In5	

FIG. 24

Lower Control Bits mux_s<4:0> mux_sp<4:0>					Higher Control Bits mux_sh<3:0> mux_shp<3:0>				cqf, aqf apf
4	3	2	1	0	3	2	1	0	Clock phase
X	X	X	X	X	0	0	0	0	Z
0	0	0	0	0	X	X	X	X	Z
0	0	0	0	1	0	0	0	1	In0
0	0	0	1	0	0	0	0	1	In1
0	0	1	0	0	0	0	0	1	In2
0	1	0	0	0	0	0	0	1	In3
1	0	0	0	0	0	0	0	1	In4
1	0	0	0	0	0	0	1	0	In5
0	1	0	0	0	0	0	1	0	In6
0	0	1	0	0	0	0	1	0	In7
0	0	0	1	0	0	0	1	0	In8
0	0	0	0	1	0	0	1	0	In9
0	0	0	0	1	0	1	0	0	In10
0	0	0	1	0	0	1	0	0	In11
0	0	1	0	0	0	1	0	0	In12
0	1	0	0	0	0	1	0	0	In13
1	0	0	0	0	0	1	0	0	In14
1	0	0	0	0	1	0	0	0	In15
0	1	0	0	0	1	0	0	0	In16
0	0	1	0	0	1	0	0	0	In17
0	0	0	1	0	1	0	0	0	In18
0	0	0	0	1	1	0	0	0	In19

FIG. 25

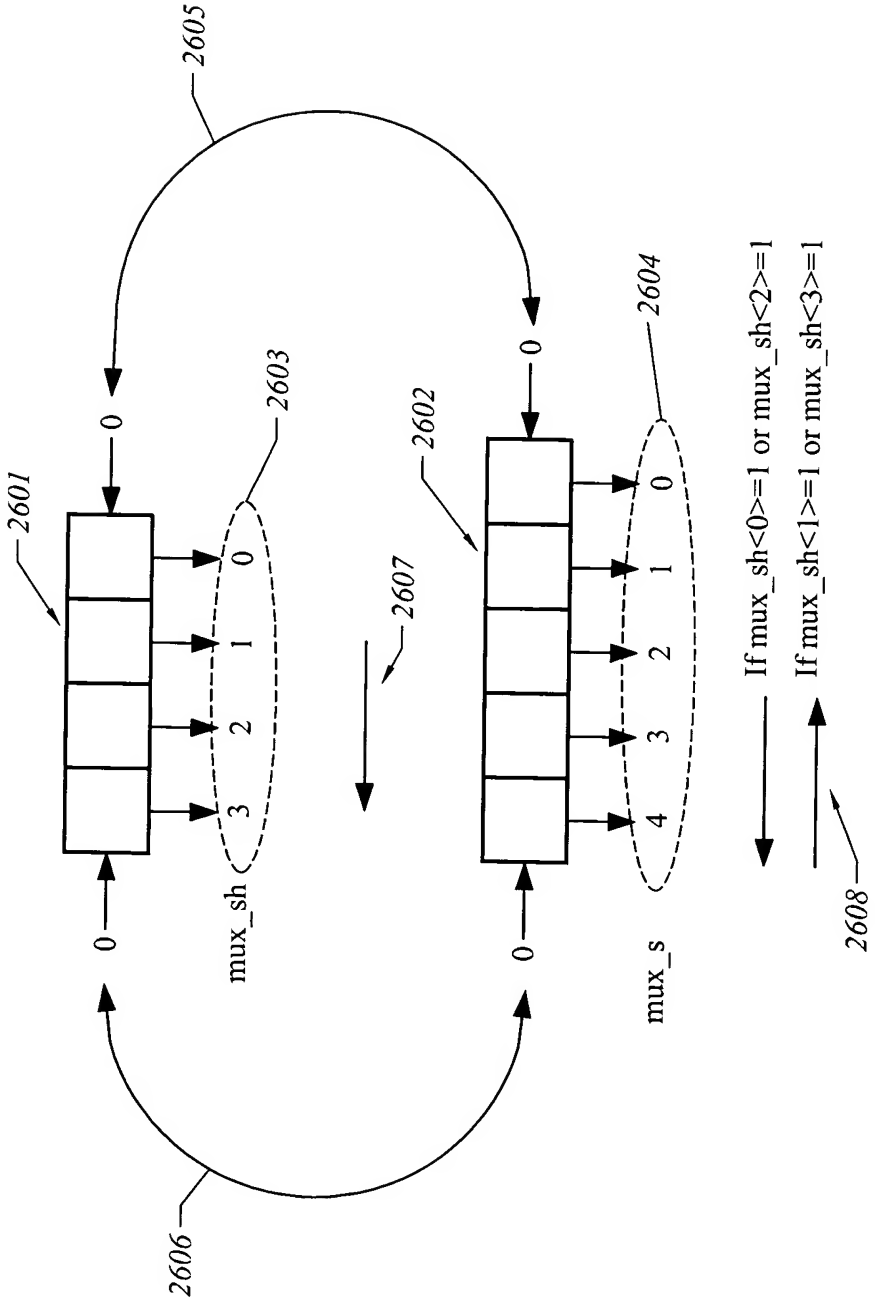

2605
2601
2603
2602
2604
2607
2606
2608
0
1
2
3
4
mux_sh
mux_s
If mux_sh<0>=1 or mux_sh<2>=1
If mux_sh<1>=1 or mux_sh<3>=1

FIG. 26